1. A pipeline microprocessor having an architected register file and at least one final execution unit for generating final results of instructions, comprising:

an address stage, located earlier in the pipeline than a stage in which the final execution unit is located, including an early execution unit, configured to generate early results of instructions prior to generation of the final results of the instructions by the final execution unit; and

an early register file, coupled to said early execution unit, corresponding to the architected register file, configured to store said early results and to provide said early results to said early execution unit for generating early results of subsequent instructions, wherein the architected register file is updated only with the final results and not with said early results.

2. The microprocessor of claim 1, wherein said early execution unit is configured to execute only a subset of the instruction set of the microprocessor.

3. The microprocessor of claim 2, wherein said subset of instructions executable by said early execution unit includes shift instructions.

4. The microprocessor of claim 2, wherein said subset of instructions executable by said early execution unit includes shift instructions and Boolean instructions.

5. The microprocessor of claim 2, wherein said subset of instructions executable by said early execution unit includes shift instructions, Boolean instructions, and arithmetic instructions.

6. The microprocessor of claim 1, further comprising:

a bus, coupled to said early execution unit, for providing a first early result of a first instruction generated by said early execution unit during a first clock cycle to said early execution unit for generating a second early result of a second instruction during a second clock cycle, wherein said first clock cycle immediately precedes said second clock cycle.

7. The microprocessor of claim 6, wherein said first instruction immediately precedes said second instruction by one pipeline stage.

8. The microprocessor of claim 1, further comprising:

a result writeback stage, located later in the pipeline than a stage in which the final execution unit is located, configured to update the architected register file with the final results generated by the final execution unit, wherein only said early register file is updated with said early results.

9. The microprocessor of claim 1, wherein said early execution logic is also configured to generate memory addresses using said early results received as operands from said early register file.

10. The microprocessor of claim 9, wherein said early execution unit is configured to generate memory addresses for stack memory locations and to generate memory addresses for non-stack memory locations.

11. The microprocessor of claim 1, wherein said early results stored in said early register file may or may not be valid.

12. The microprocessor of claim 1, wherein the pipeline microprocessor is a scalar microprocessor.

13. The microprocessor of claim 1, wherein the pipeline microprocessor issues instructions in program order.

14. The microprocessor of claim 1, wherein said early results generated by said early execution unit comprise non-address results and stack pointer addresses, and said early execution unit resides in a single pipeline stage.

15. The microprocessor of claim 1, wherein a computer program product comprising a computer usable medium having computer readable program code causes the microprocessor, wherein said computer program product is for use with a computing device.

16. The microprocessor of claim 1, wherein a computer data signal embodied in a transmission medium comprising computer-readable program code provides the microprocessor.

17. An apparatus for improving performance in a pipeline microprocessor having an architected register file, the apparatus comprising:

early execution logic, configured to generate early results of instructions prior to generation of final results of said instructions by one or more execution units lower in the microprocessor pipeline than said early execution logic;

an early register file, coupled to said early execution logic, comprising registers corresponding to the architected register file for storing said early results, having a valid indicator associated with each of said registers, configured to provide said early results to said early execution logic as operands for generating subsequent said early results; and

logic, coupled to receive said valid indicators, configured to stall the pipeline if said early execution logic is generating an address using an operand stored in one of said early register file registers and said valid indicators indicate that said register is invalid, wherein said logic is further configured not to stall the pipeline if said early execution logic is generating a non-memory address early result using an operand stored in one of said early register file registers and said valid indicators indicate that said register is invalid.

18. The apparatus of claim 17, wherein said early execution logic is within an address stage of the pipeline microprocessor.

19. The apparatus of claim 17, further comprising:

a register, coupled to said logic, configured to accumulate early status flags based on said early results, thereby enabling early execution of conditional instructions based on said early status flags accumulated in said register.

20. The apparatus of claim 17, further comprising:

a result bus, coupling an output of said early execution logic to a first input of a multiplexer, for providing said early results from said early execution logic, said multiplexer having a second input coupled to receive an operand from the architected register file, said multiplexer having an output for providing at least one operand selected from at least one of said inputs to said early execution logic, whereby a first early result generated by said early execution logic during a first clock cycle is provided as an input operand to a second instruction for generating a second early result of said second instruction during a second clock cycle immediately subsequent to said first clock cycle.

21. The apparatus of claim 17, wherein the pipeline microprocessor is a scalar microprocessor.

22. An apparatus in a pipeline microprocessor for generating memory addresses and early instruction results, the apparatus comprising:

an address generator, configured to generate a memory address if an instruction specifies memory address generation, and further configured to generate a selectively valid result if said instruction specifies result generation; and

logic, coupled to said address generator, configured to stall the pipeline if said instruction specifies memory address generation and not all input operands of said instruction are valid and available for provision to said address generator, and further configured not to stall the pipeline if said instruction specifies result generation and not all input operands of said instruction are valid and available for provision to said address generator.

23. The apparatus of claim 22, wherein said address generator generates memory addresses for stack memory locations and generates memory addresses for non-stack memory locations.

24. The apparatus of claim 22, wherein said address generator resides in a single pipeline stage.

25. The apparatus of claim 22, wherein said logic generates an indicator to indicate said result is invalid if said address generator is not configured to perform a type of operation specified by said instruction.

26. The apparatus of claim 22, wherein said logic generates an indicator to indicate said result is invalid if invalid source operands are provided to said address generator to generate said result.

27. The apparatus of claim 22, wherein the pipeline microprocessor is a scalar microprocessor.

28. A method for avoiding stalls in a microprocessor pipeline caused by operand dependencies between instructions, the microprocessor having an architected register file, the method comprising:

generating a first result of a first instruction in an address stage of the pipeline, wherein the first result is potentially invalid;

storing the first result into an early register file of the microprocessor;

generating a second result of a second instruction in the address stage using the first result from the early register file as an input operand to the second instruction;

storing the second result into the early register file; and

generating a memory address for a third instruction using the second result from the early register file as an input operand to the third instruction.

29. The method of claim 28, further comprising:

accumulating early status flags in response to the second result.

30. The method of claim 29, further comprising:

invalidating the early status flags if the second result is invalid.

31. The method of claim 30, further comprising:

determining whether a condition specified by a conditional instruction is satisfied by the early status flags; and

performing an operation specified by the conditional instruction if the early status flags are valid and the condition is satisfied in the early status flags.

32. A computer data signal embodied in a transmission medium, comprising:

computer-readable program code for providing a pipeline microprocessor having an architected register file and at least one final execution unit for generating final results of instructions, said program code comprising:

first program code for providing an address stage, located earlier in the pipeline than a stage in which the final execution unit is located, including an early execution unit, configured to generate early results of instructions prior to generation of the final results of the instructions by the final execution unit; and

second program code for providing an early register file, coupled to said early execution unit, corresponding to the architected register file, configured to store said early results and to provide said early results to said early execution unit for generating early results of subsequent instructions, wherein the architected register file is updated only with the final results and not with said early results.